

**LAPORAN PENELITIAN KELOMPOK**

**PEMBANGKITAN DAN OPTIMALISASI  
JADWAL KULIAH DENGAN MENGGUNAKAN  
METODE *PARTICLE SWARM OPTIMIZATION* (PSO)**



Disusun oleh:  
Paulus Mudjihartono, S.T., M.T.

Fakultas Teknologi Industri  
Universitas Atma Jaya Yogyakarta  
2011

## LEMBAR PENGESAHAN LAPORAN PENELITIAN

No. Laporan : \_\_\_\_\_

- 1.a) Judul Penelitian : Pembangkitan Dan Optimalisasi Jadwal Kuliah Dengan Menggunakan Metode *Particle Swarm Optimization* (PSO)
- b) Macam Penelitian : Laboratorium
2. Peneliti
  - a) Nama : Paulus Mudjihartono, ST., M.T.
  - b) Jenis Kelamin : Laki-laki
  - c) Usia saat pengajuan proposal : 41 Tahun 7 bulan
  - d) Jabatan akademik, golongan : Lektor Kepala, III/d
  - e) Bidang peminatan : *Enterprise Information System*
  - f) Fakultas, Program Studi : Teknologi Industri, Teknik Informatika
3. Jangka waktu penelitian : 6 bulan
4. Biaya yang disetujui : Rp. 3.065.000

Yogyakarta, 28 Februari 2011

Peneliti,

Paulus Mudjihartono, ST., M.T.

Mengetahui,  
Kepala Laboratorium Struktur Data FTI

Eduard Rusdianto, S.T., M.T.

Dekan Fakultas Teknologi Industri UAJY,                      Ketua LPPM UAJY,

Ir. B Kristyanto, M.Eng., Ph.D

Dr. MF Shellyana Junaedi, SE., M.Si

## INTISARI

Salah satu masalah yang harus dipecahkan oleh seorang pejabat akademik di universitas (kepala program studi) adalah membuat jadwal kuliah. Ada banyak kombinasi jadwal yang dapat dibuat dari sekelompok matakuliah dan dosen. Namun, tidak semua kombinasi jadwal tersebut memuaskan. Dalam jadwal kuliah ada kendala (*constraint*) yang harus dipenuhi. Jika jadwal tidak memenuhi kendala, maka jadwal tersebut ditolak. Dengan demikian, membuat jadwal kuliah dapat dipandang sebagai masalah pencarian (*searching*) kombinasi matakuliah-dosen yang memenuhi kendala. Pencarian jadwal kuliah yang optimal (yang paling tidak melanggar kendala, akan lebih baik lagi tidak melanggar) perlu dipecahkan. Ada beberapa usaha telah dilakukan, misalnya dengan menggunakan metode pewarnaan graf, dan metode *greedy*.

Jika kendala didekati dengan nilai *penalty* (hukuman) maka masalah dapat dipandang sebagai bagaimana meminimalkan *penalty* tersebut. Oleh karena, masalah pembangkitan jadwal ini, akhirnya dapat dipandang juga sebagai masalah optimalisasi, yaitu bagaimana memaksimalkan nilai yang selaras dengan keinginan dan meminimalkan nilai *penalty* pada saat yang bersamaan. PSO menawarkan solusi secara heuristik untuk mencari nilai optimal ini. Yang dilakukan hanya memetakan masalah pembangkitan jadwal kuliah ini ke dalam ranah masalah PSO. Fungsi objektif dan kendala perlu ditentukan dalam konteks PSO.

Hasil penelitian menunjukkan jadwal kuliah yang dibangkitkan optimal (memenuhi kendala). Namun demikian, jika kendala diubah menjadi ekstrem, seperti ruang atau sesi dikurangi maka jadwal yang terbangkitkan tidak optimal. Optimalnya jadwal dapat juga menunjukkan kurangnya sumber daya jadwal, seperti ruang atau sesi.

**Keyword:** jadwal kuliah, kendala jadwal, pemetaan masalah, PSO.

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan yang maha Esa karena atas berkatNya, akhirnya penulis dapat menyelesaikan penelitian dengan judul "Pembangkitan Dan Optimalisasi Jadwal Kuliah Dengan Menggunakan Metode *Particle Swarm Optimization* (PSO)" yang diajukan sebagai penelitian di Lembaga Penelitian Universitas Atma Jaya Yogyakarta.

Dalam kesempatan ini penulis mengucapkan terimakasih kepada:

1. Dr. MF. Shellyana Junaedi, SE., M.Si, selaku Ketua Lembaga Penelitian dan Pengabdian pada Masyarakat (LPPM) Universitas Atma Jaya Yogyakarta.
2. Ir. B. Kristyanto, M.Eng, Ph.D, selaku Dekan Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta.
3. Ir. Suyoto, M.Sc., Ph.D., selaku Ketua Program Studi Teknik Informatika, Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta.
4. Eduard Rusdianto, S.T., M.T., selaku Kepala Laboratorium Struktur Data Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta.
5. Rekan-rekan di Fakultas Teknologi Industri UAJY yang tidak dapat disebutkan satu persatu.

Tak lupa penulis mohon masukan yang bersifat korektif agar tulisan ini dapat menjadi lebih baik. Akhir kata, semoga tulisan ini dapat bermanfaat bagi kita semua.

Yogyakarta, Februari 2011

Peneliti

## DAFTAR ISI

LEMBAR PENGESAHAN LAPORAN PENELITIAN .....	ii
INTISARI .....	iii
KATA PENGANTAR .....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR .....	vi
DAFTAR TABEL.....	vii
1. BAB I LATAR BELAKANG .....	1
2. BAB II TINJAUAN PUSTAKA .....	3
2.1. Solusi dengan Pewarnaan Graf .....	3
2.2. Metode Taboo Search .....	3
3.1. Perumusan Masalah .....	5
3.2. Tujuan Penelitian .....	5
3.3. Manfaat Masalah .....	5
3. BAB IV METODOLOGI PENELITIAN .....	6
4.1. Analisis Kebutuhan Perangkat Lunak .....	6
4.2. Perancangan Kebutuhan Perangkat Lunak .....	10
4.3. Pengkodean Perangkat Lunak .....	12
4. BAB V HASIL DAN PEMBAHASAN .....	21
5. BAB VI KESIMPULAN DAN SARAN .....	24
6.1. Kesimpulan .....	24
6.2. Saran .....	24
DAFTAR PUSTAKA .....	25

## DAFTAR GAMBAR

Gambar 1. Template Jadwal Kuliah yang Dikehendaki-----	6
Gambar 2. Use Case Diagram Pembangkitan jadwal Kuliah-----	7
Gambar 3. Mekanisme Balik dari Partikel ke Jadwal Kuliah -----	9
Gambar 4. Contoh Pemetaan Balik Partikel ke Jadwal-----	10
Gambar 5. Entity Relationship Diagram (ERD) Pembangkitan Jadwal Kuliah---	10
Gambar 6. Sequence Diagram Pembangkitan Jadwal -----	12
Gambar 7. Diagram Kelas Pembangkitan Jadwal Kuliah-----	13
Gambar 8. Antarmuka Pembangkitan Jadwal Kuliah-----	15
Gambar 9. Antarmuka Cetak Jadwal -----	17
Gambar 10. Antarmuka History -----	19

## DAFTAR TABEL

Tabel 1. Pengujian Fungsional Jadwal Pembangkitan Kuliah -----	21
Tabel 2. Pengujian Fungsional Penyimpanan Hasil Jadwal Kuliah -----	21
Tabel 3. Pengujian Fungsional Proses Menampilkan Hasil Jadwal Kuliah-----	22



## **BAB I**

### **LATAR BELAKANG**

Dalam tugas-tugas rutin akademik, seorang pejabat akademik membutuhkan membuat jadwal kuliah yang dapat memenuhi semua keinginan dan mengatasi semua keterbatasan. Keinginan-keinginan atau preferensi ini dapat muncul dari beberapa pihak, misalnya dari dosen yang mempunyai preferensi hari tertentu, preferensi mahasiswa tentang jam kuliah (sesi). Sedangkan keterbatasan dapat ditimbulkan dari sumber daya dan regulasi. Keterbatasan ini biasanya bersifat *mandatory*, misalnya ketersediaan ruangan kelas. Tidak dapat dibuat jadwal yang ruangnya tidak ada. Sumber daya lain yang membatasi misalnya adalah slot waktu. Sumber daya adalah keterbatasan yang *given* secara alami sehingga tidak dapat ditawar. Regulasi merupakan keterbatasan lain yang sifatnya *compelling* (memaksa). Secara matematis dapat dibuat jadwal yang melanggar regulasi, tetapi tentu saja tidak dikehendaki. Dengan kata lain, pembangkitan jadwal adalah masalah mencari jadwal yang memenuhi kendala-kendala yang ada. Jika semua faktor yang memenuhi kendala diberi nilai positif dan yang melanggar kendala diberi nilai negatif, maka pembangkitan jadwal setara dengan masalah optimalisasi, yaitu mencari nilai maksimal dari kombinasi jadwal yang mungkin dibuat.

*Particle Swarm Optimization* (PSO) menawarkan solusi secara heuristik. PSO memperkenalkan partikel atau individu sebagai sebuah solusi (kandidat solusi) yang akan terus dicek validitasnya (Engelbrecht, 2007). Dalam PSO terdapat swarm atau populasi, yaitu kumpulan partikel yang masing-masing merepresentasikan solusi. Sebuah partikel bergerak dalam ruang pencarian menuju ke keadaan optimal oleh karena dua mekanisme, yaitu (1) arah dengan besar acak yang mempertimbangkan keadaan terbaik partikel tersebut, dan (2) arah dengan besar acak yang mengarah ke keadaan terbaik secara global (menurut semua partikel). Posisi partikel dan kecepatannya dimutakhirkan setiap saat sampai seluruh populasi konvergen (Chan & Tiwari, 2008).



PSO ini merupakan metode optimasi stokastik berdasarkan pada konsep populasi (sekumpulan burung, ikan, lebah, dll) yang bisa digunakan untuk menyelesaikan permasalahan non linier dan memiliki berbagai keunggulan yaitu lebih cepatnya mencapai konvergensi bila dibandingkan dengan metode heuristik yang lain. PSO diinisialisasi dengan sebuah populasi dari solusi-solusi acak dan mencari solusi yang paling optimal dengan membaharui anggota populasi (Wahyu, 2010).

Permasalahan optimalisasi jadwal kuliah dapat dipecahkan dengan PSO dengan cukup baik. Yang dilakukan adalah memetakan permasalahan optimalisasi tersebut ke dalam ranah masalah PSO. Sebuah tabel jadwal dapat dianggap sebagai sebuah partikel. Sebuah partikel ditentukan posisinya oleh koordinat-koordinat penyusunnya yang dalam jadwal kuliah direpresentasikan dengan urutan matakuliah diinsertkan ke slot ruang-sesi. Bergeraknya partikel ini ke posisi yang lain, direpresentasikan dengan pemutakhiran urutan matakuliah disisipkan ke slot ruang-sesi. Posisi terakhir partikel inilah jawaban dari permasalahan optimalisasi jadwal ini.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Solusi dengan Pewarnaan Graf**

Beberapa solusi pembangkitan jadwal telah banyak ditempuh oleh beberapa peneliti terdahulu. Dewi (2006) mengusulkan solusi dengan menggunakan pewarnaan graf. Metode pewarnaan graf didasarkan pada karakteristik graf dengan tambahan aturan. Aturannya ialah: setiap sisi (*edge*) dalam graf diberi warna. Macamnya warna diusahakan sesedikit mungkin dengan aturan, warna sisi hanya boleh sama jika tidak mempunyai simpul (*vertex*) yang sama. Sisi yang tidak boleh sama warna merepresentasikan kendala dalam jadwal kuliah, seperti dua matakuliah dari semester yang sama tidak boleh dijadwalkan dalam hari dan sesi yang sama. Kendala lain yang setara dengan ini ialah: dua matakuliah yang sama yang diajar satu dosen tidak dapat dijadwalkan dalam hari dan sesi yang sama. Selama kendala tersebut dipenuhi, maka sebuah solusi akan tercipta. Sayangnya, tujuan optimalisasi ini adalah hanya meminimalkan sumberdaya saja. Kendala lain seperti preferensi tidak dipertimbangkan.

#### **2.2. Metode Taboo Search**

Penelitian lain menampilkan optimalisasi sejenis dengan metode *taboo search*. Metode ini mengasumsikan slot waktu yang seragam, dan mencoba mengepaskan ruang kelas untuk cukup bagi dosen. Di metode ini, Mieke et al (2006) mengusulkan adanya agregasi sesi. Agregasi sesi, disebut *pillar* yang berisi sekumpulan sesi untuk sebuah matakuliah. Jadi, setiap matakuliah dapat saja diwujudkan dalam sejumlah sesi yang berbeda, atau dengan kata lain *pillar* yang ada berbeda. Hal ini dikerjakan karena memang sifat jadwal kuliah yang hendak dipecahkan berbasis *pillar*. Tentu saja hal ini tidak sesuai dengan banyak kasus di tempat lain, seperti di program studi Teknik Informatika Universitas Atma Jaya Yogyakarta, yang mendasarkan pada sesi yang seragam, sehingga tidak diperlukan *pillar* atau kumpulan sesi.

### **2.3. Metode Berbasis Algoritma Genetik (*Genetic Algorithm*)**

Penelitian Wang (2009) menggunakan pendekatan berbasis algoritma genetik, berhasil membangkitkan jadwal kuliah bagi mahasiswa dan dosen. Algoritma yang ditempuh mengadopsi kemampuan swa-subur dari beberapa tumbuhan tertentu. Pemetaan masalah tetap diperlukan, yaitu dengan mentransformasikan fungsi objektif, seleksi dan *cross-over* menjadi yang sesuai dengan permasalahan pembangkitan jadwal. Hasil yang didapat, jadwal kuliah dapat dibangkitkan dengan konvergensi yang cepat.

### **2.4. Particle Swarm Optimization (PSO)**

Metode PSO telah banyak diterapkan untuk mengatasi jadwal matakuliah. Masalah jadwal matakuliah yang dipecahkan PSO ini sering disebut sebagai *University Timetable Problem* (UTP). Namun tidak semua solusi mempunyai fungsi objektif dan kendala yang sama. Beberapa solusi yang ditawarkan berbeda asumsi satu dengan yang lainnya. Misalnya Fen et al (2009) menggunakan hybrid PSO, yaitu kombinasi PSO *local search* dan pendalihan (*reasoning*) berbasis kendala. Hasil yang diperoleh menunjukkan jadwal matakuliah yang lebih baik.

### **BAB III**

#### **MASALAH, TUJUAN DAN MANFAAT PENELITIAN**

##### **3.1. Perumusan Masalah**

Dari uraian latar belakang masalah diatas maka dapat dirumuskan permasalahan yang ada:

Bagaimana membangkitkan jadwal matakuliah yang tunduk pada beberapa kendala (seperti *free-collision*, preferensi sesi, minimalisasi sumber daya) dengan metode *particle swarm optimization*?

##### **3.2. Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah:

Memperoleh jadwal mata kuliah yang diperoleh dari kombinasi optimal untuk pasangan matakuliah dan dosen secara keseluruhan, tidak ada permasalahan bentrokan jadwal (*free-collision*) pada sisi mahasiswa, serta ketersediaan ruang yang cukup. Pembangkitan jadwal matakuliah ini menggunakan metode PSO.

##### **3.3. Manfaat Masalah**

Manfaat penelitian ini secara umum, yaitu:

1. Mampu memberikan jadwal kuliah secara cukup optimal (asal tidak melanggar kendala) tetapi dengan waktu yang cepat.
2. Mampu memberikan informasi mengenai sumber daya yang tersedia, baik cukup maupun kurang.

## BAB IV

### METODOLOGI PENELITIAN

Metode Penelitian yang dilakukan adalah dengan mengadopsi metodologi pembangunan perangkat lunak. Metodologi ini meliputi tahapan sebagai berikut: (1) analisis kebutuhan perangkat lunak (2) perancangan perangkat lunak (3) implementasi perancangan perangkat lunak dan (4) pengujian perangkat lunak.

#### 4.1. Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan dimulai dengan studi dokumen, yaitu mencari bentuk jadwal kuliah yang berlaku di program studi Teknik Informatika, Universitas Atma Jaya Yogyakarta. Jadwal kuliah yang diinginkan dapat ditunjukkan pada gambar 1 di bawah ini.

Hari	Sesi	Ruang 1	Ruang 2	Ruang 3	.....
Senin	sesi 1	kuliah1			
	sesi 2	kuliah2	kuliah3		
	sesi 3				
	sesi 4				
	sesi 5			kuliah4	
Selasa	sesi 1				
	sesi 2				kuliah.. n
	sesi 3				
	sesi 4				
	sesi 5				
.....					.....

Gambar 1. Template Jadwal Kuliah yang Dikehendaki

Kebutuhan yang dipersyaratkan terkategori ke dalam dua kebutuhan, yaitu yang disebabkan oleh sifat alami jadwal dan oleh preferensi. Kebutuhan yang disebabkan sifat alami jadwal bersifat *mandatory* atau wajib dipenuhi. Sedangkan kebutuhan preferensi bersifat opsional.

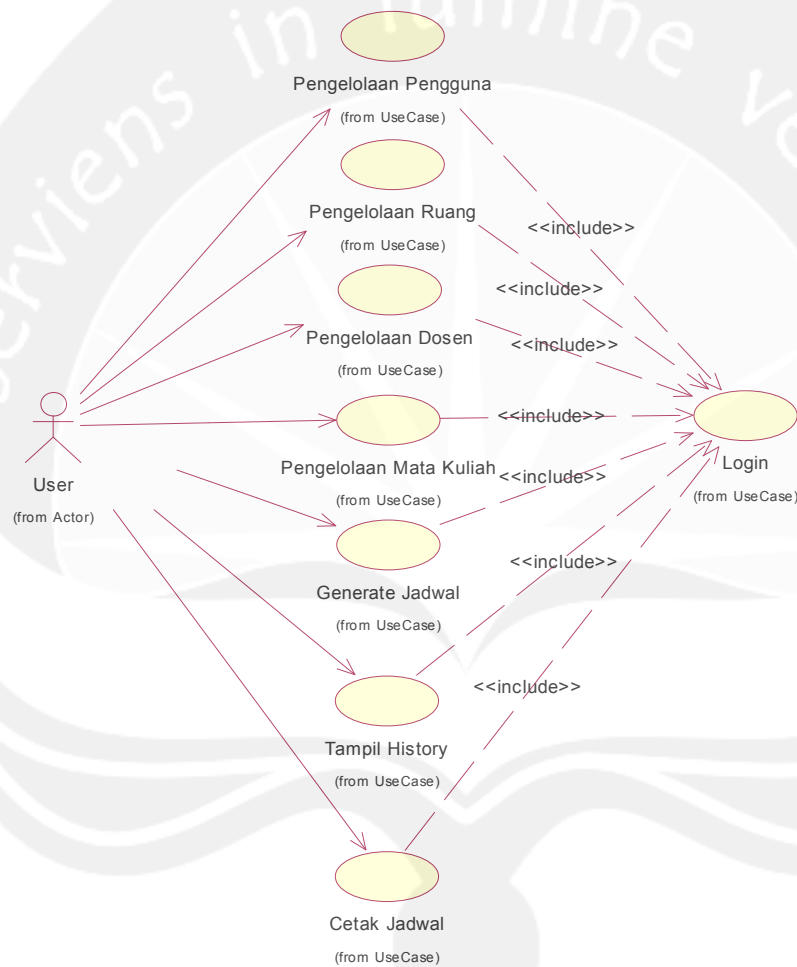
Kebutuhan oleh sifat alami jadwal kuliah:

1. Tidak ada dua atau lebih jadwal kuliah dengan dosen yang sama diselenggarakan dengan hari dan sesi (slot waktu) yang sama.
2. Tidak ada solusi untuk kasus banyaknya sel jadwal-matakuliah lebih banyak daripada slot waktu-ruang.

Kebutuhan oleh preferensi:

1. Tidak ada dua atau lebih matakuliah dari semester yang sama dengan hanya satu kelas diselenggarakan dengan slot waktu yang sama.
2. Jadwal dapat merefer ke sesi tertentu.
3. Diusahakan matakuliah-matakuliah dari semester yang sama dengan banyak kelas dapat diselenggarakan dalam banyak alternatif pengambilan bagi mahasiswa.

Dalam pendekatan paradigma objek, kebutuhan perangkat lunak ini dapat digambarkan dalam sebuah diagram use case, seperti pada gambar 2 di bawah ini.



Gambar 2. Use Case Diagram Pembangkitan Jadwal Kuliah

Pemetaan masalah dari ranah masalah orisinal ke ranah masalah PSO, selanjutnya perlu ditempuh. Pemetaan ini akan menyiapkan bentuk fungsi objektif dan kendala-kendala yang sesuai dengan permasalahan PSO. Ada dua pemetaan yang dilakukan, yaitu:

**Pemetaan solusi:**

(1) Sebuah jadwal kuliah dianggap sebagai sebuah solusi yang dipetakan sebagai sebuah partikel dalam PSO.

(2) Sebuah sel sesi-ruang menunjukkan sebuah dimensi partikel. Jika ada  $n$  sel maka partikel terletak dalam ruang pencarian (search-space) yang berdimensi  $n$ .

(3) Urutan kuliah dimasukkan ke dalam sel sesi-ruang memetakan posisi partikel suatu saat tertentu pada ruang pencarian. Setiap sel mempunyai nilai yang dapat dipandang sebagai sebuah nilai dari sebuah dimensi. Urutan memasukkan kuliah ke dalam jadwal akan diikuti dengan penciptaan posisi partikel (solusi) untuk jadwal tersebut. Penciptaan posisi ini dengan sederhana dapat ditempuh dengan mencatat setiap nilai sel setiap kali kuliah diinputkan, maka dengan selesainya semua kuliah diinputkan ke jadwal, akan tercipta sederet nilai (bilangan) sebanyak  $n$  bilangan. Deret bilangan inilah yang akan dianggap sebagai vektor posisi dengan  $n$  dimensi. Alogortima yang sama diterapkan untuk solusi-solusi lain.

(4) Perubahan atau pemutakhiran solusi ke dalam solusi yang lebih baik dipetakan sebagai pergerakan partikel dalam ruang pencarian.

(5) Fungsi objektifnya adalah bagaimana meminimalkan nilai jadwal kuliah. Dengan asumsi semua yang memenuhi kendala jadwal dinilai positif dan yang melanggar kendala dinilai negatif.

**Pemetaan Balik dari Partikel kembali ke Jadwal**

(1) Tandai setiap kuliah (matakuliah-kelas) dengan sebuah integer unik.

(2) Tandai sel dengan integer unik juga.

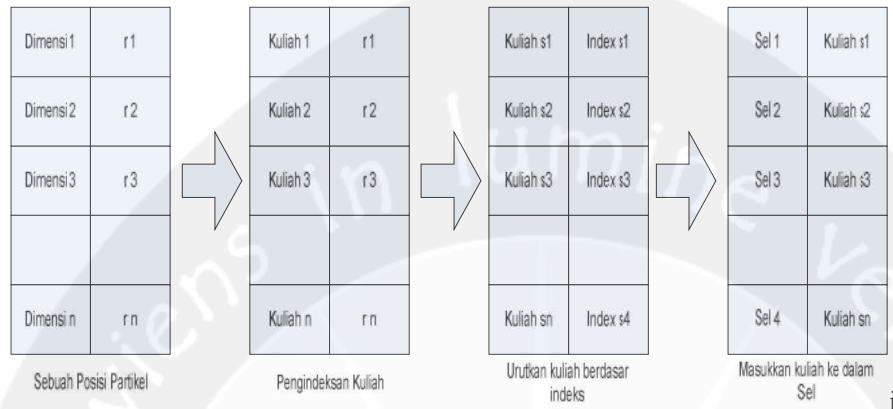
(3) Tandai lagi setiap kuliah dengan bilangan real acak unik. Angka ini merepresentasikan posisi partikel/solusi dalam ruang pencarian

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

*hyperspace*. Namai bilangan ini sebagai indeks kuliah (setara dimensi dalam partiel).

- (4) Urutkan kuliah menurut indeksnya. Masukkan ke jadwal sesuai urutan tersebut. Dengan demikian akan tercipta jadwal/solusi baru. (lihat gambar 3)



Gambar 3. Mekanisme Balik dari Partikel ke Jadwal Kuliah

Lebih lanjut, dalam kasus prodi teknik informatika UAJY, disampel, diperoleh pemetaan sebagai berikut:

*Particle ke i*

dimensi	1	2	3	4	5	6	7	8	9	10
nilai	0.28	0.50	0.64	0.95	0.54	0.52	0.29	0.53	0.56	0.34

↓

List Jadwal

Indeks_mk	Nama Mata Kuliah
1	Kalkulus
2	Logika Matematis
3	Fisika Elektronika
4	Statistika
5	Matematika Diskret
6	Struktur Data
7	Komputasi lanjut
8	Sistem Operasi
9	Dasar Multimedia
10	Basis Data

↓



0.28	0.50	0.64	0.95	0.54	0.52	0.29	0.53	0.56	0.34
1	2	3	4	5	6	7	8	9	10

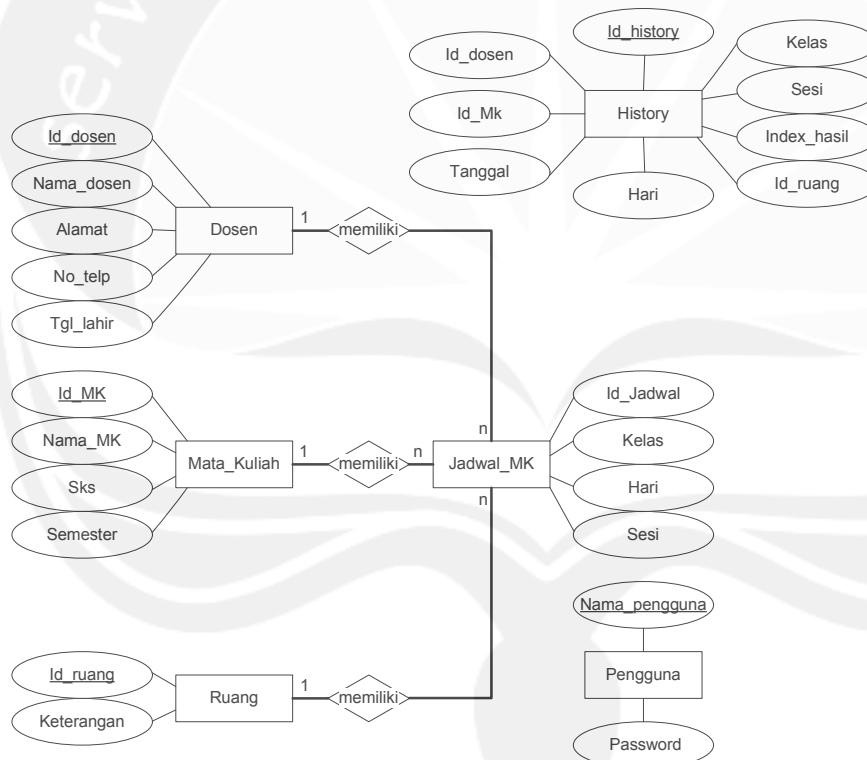


0.28	0.29	0.34	0.50	0.52	0.53	0.54	0.56	0.64	0.95
1	7	10	2	6	8	5	9	3	4

Gambar 4. Contoh Pemetaan balik Partikel ke Jadwal

#### 4.2. Perancangan Kebutuhan Perangkat Lunak

Pada perancangan dilakukan perancangan pemodelan data, prosedural dan antarmuka. Pemodelan data dapat dengan mudah diciptakan, karena perangkat lunak ini ditekankan lebih sebagai piranti bantu daripada sistem itu sendiri. Beberapa entitas diciptakan dengan sederhana seperti dosen, matakuliah dan ruang. Jadwal menampung primary-key ketiga entitas ini sebagai foreign-key. Diagram relasi entitas ditunjukkan dalam gambar di bawah ini.



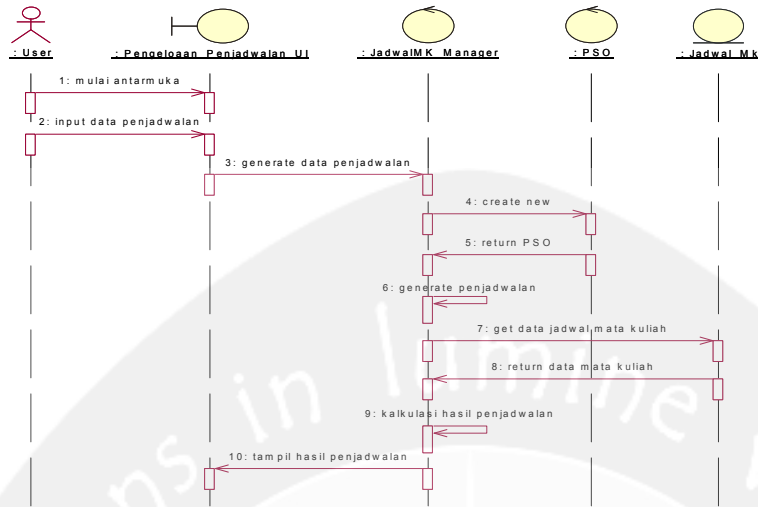
Gambar 5. Entity Relationship Duagram (ERD) Pembangkitan Jadwal Kuliah

Perangkat lunak yang dibutuhkan dalam mengoperasikan perangkat lunak  
Pembangkitan Jadwal Kuliah ini adalah sebagai berikut:

1. Nama : Windows XP atau lebih tinggi  
Sumber : Microsoft Corporation  
Sebagai sistem operasi dimana perangkat lunak AUT dijalankan.
2. Nama : SQL Server 2000  
Nomor Versi : 8.0  
Sumber : Microsoft Corporation  
Sebagai DBMS yang dibutuhkan dalam mengoperasikan perangkat lunak AUT.
3. Nama : .Net  
Nomor Versi : 2.0  
Sumber : Microsoft Corporation  
Sebagai platform sistem yang menjalankan perangkat lunak AUT.

### ***Sequence Diagram***

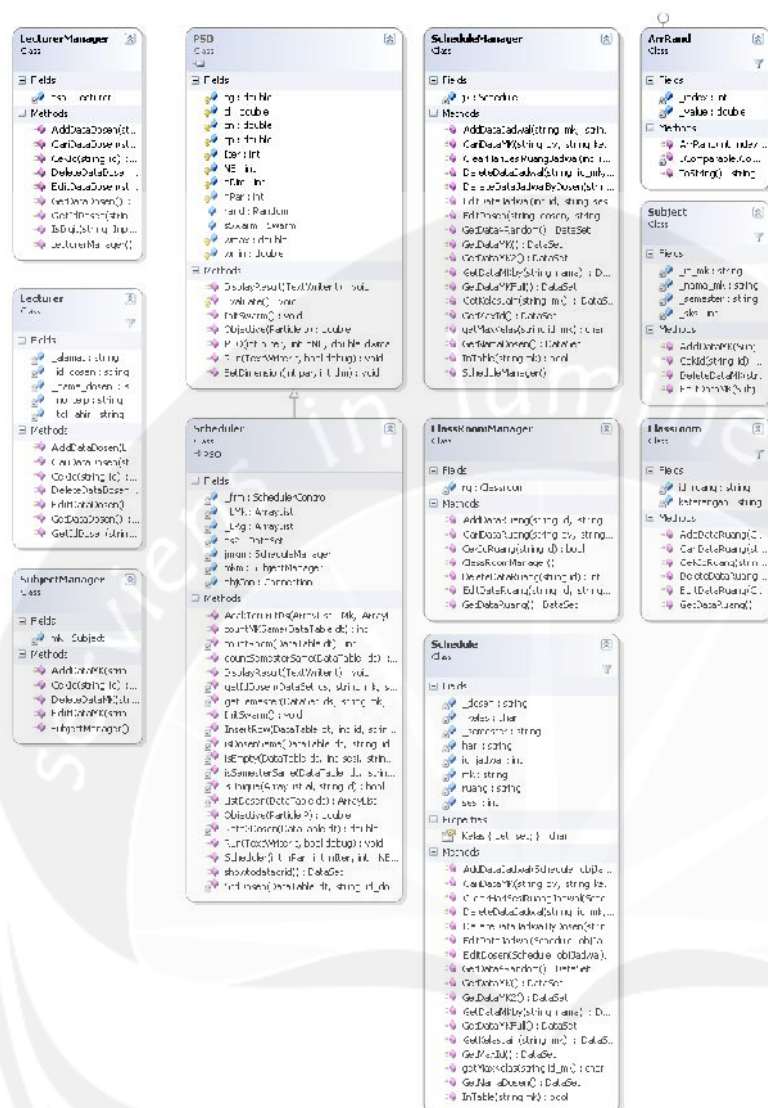
Untuk *Sequence Diagram* (SD) dipergunakan untuk menunjukka aspek dinamis dari perangkat lunak. SD dapat dilihat pada gambar 4 di bawah ini:



Gambar 6. Sequence Diagram Pembangkitan Jadwal

#### 4.3. Pengkodean Perangkat Lunak

Pada pengkodean diciptakan kelas-kelas yang akan dipergunakan dalam eksekusi perangkat lunak pembangkitan jadwal kuliah ini. Kelas-kelas pendukung dirangkum dalam kelas diagram di bawah ini:



Gambar 7. Diagram Kelas Pembangunan Jadwal Kuliah

Adapun realisasi PSO ditunjukkan dengan algoritma di bawah ini:

Algoritma sebuah partikel bergerak ke posisi yang lebih baik ditunjukkan dalam langkah-langkah di bawah ini:

- (1) Inisialisasi: bangkitkan posisi acak dari  $m$  partikel. Setiap partikel terletak dalam ruang dimensi  $n$ .
- (2) Transformasikan setiap partikel ke sebuah jadwal kuliah. Kemudian hitung fungsi objektifnya dari setiap jadwal, isi nilai objektif partikel dengan nilai fungsi objektif milik jadwal.
- (3) Gunakan mekanisme PSO untuk memutakhirkan posisi personal best dan posisi global best-nya. Kemudian, pindahkan partikel ke posisi baru dengan sebelumnya memutakhirkan *velocity* personal and global-nya.

Gunaka rumus:

$$x_i(t+1) = x(t) + v(t+1)$$

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x(t)]_{ij}$$

dengan

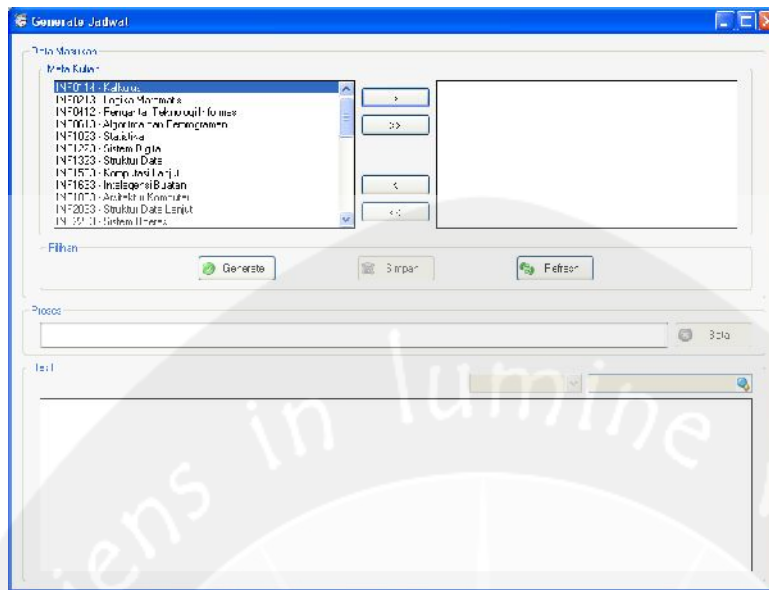
$x_i(t)$  posisi pada iterasi ke  $t$

$v(t+1)$  kecepatan pada iterasi ke  $t+1$

$c$  dan  $r$  adalah konstanta dan acak (yang akan memberikan bobot antara pengetahuan personal dan sosial setiap partikel)

- (4) kembali ke langkah (2) sampai beberapa iterasi yang dibutuhkan.
- (5) Transformasikan posisi global best sebagai jadwal terbaik yang ditemukan pada iterasi terakhir.

Perancangan juga meliputi antar muka perangkat lunak. Antarmuka ini digunakan oleh pengguna untuk menyusun mata kuliah menjadi sebuah jadwal mata kuliah menggunakan metode PSO.



Gambar 8 Antarmuka Pembangkitan Jadwal Kuliah

Fungsi yang dapat dilakukan melalui antarmuka ini adalah: generate jadwal, simpan jadwal, refresh, tampil penjadwalan dan cari jadwal.

Generate jadwal digunakan untuk menyusun mata kuliah menjadi sebuah jadwal mata kuliah, simpan jadwal digunakan untuk menyimpan hasil penyusunan jadwal ke dalam basis data, refresh digunakan untuk mengkondisikan antarmuka seperti keadaan waktu pertama kali dibuka, dan cari jadwal digunakan untuk mencari data jadwal di dalam tabel jadwal.

Data harus dipilih terlebih dahulu sebelum melakukan generate jadwal, sedangkan untuk melakukan simpan jadwal dan cari jadwal, pengguna diharuskan memilih melakukan proses generate jadwal terlebih dahulu.

#### On Form Load

Tabel jadwal akan dikosongkan dan listbox mata kuliah akan diisi id dan nama mata kuliah dari tabel mata kuliah yang telah memiliki dosen pengajar.

#### On Click Button Generate

Akan dilakukan pengecekan apakah data yang dipilih telah benar. Jika sudah, maka akan dilakukan penyusunan mata kuliah menjadi sebuah jadwal mata kuliah. Selama proses berlangsung, progressbar akan menambahkan nilainya

dengan sendirinya, sehingga apabila progressbar telah penuh, berarti proses telah selesai. Kemudian hasil dari penyusunan jadwal tersebut akan ditampilkan di tabel jadwal. Jika belum maka akan ditampilkan pesan “Pilih data terlebih dahulu”

#### **On Click Button Simpan**

Akan dilakukan penyimpanan ke dalam basis data, hasil penyusunan jadwal mata kuliah yang terdapat di dalam tabel jadwal.

*Query* yang digunakan:

“insert into history (id\_history, index\_hasil, id\_mk, hari, sesi, id\_ruang, id\_dosen, tanggal) values (?, ?, ?, ?, ?, ?, ?, ?)”.

dan

“UPDATE jadwal\_mk set id\_jadwal = ?, hari = ?, sesi = ?, id\_ruang = ?, id\_dosen = ? where id\_mk = ?”.

dan

*Query* yang digunakan:

“UPDATE jadwal\_mk set id\_jadwal = ?, hari = null, sesi = null, id\_ruang = null where nama\_pengguna = ?”.

#### **On Click Button Batal**

Akan dilakukan pembatalan proses penyusunan jadwal mata kuliah.

#### **On Click Button Refresh**

Data pada listbox mata kuliah akan dikembalikan seperti ketika pertama kali masuk ke antarmuka ini, dimana listbox kanan akan kosong dan listbox kiri akan berisi data mata kuliah yang telah memiliki dosen pengajar.

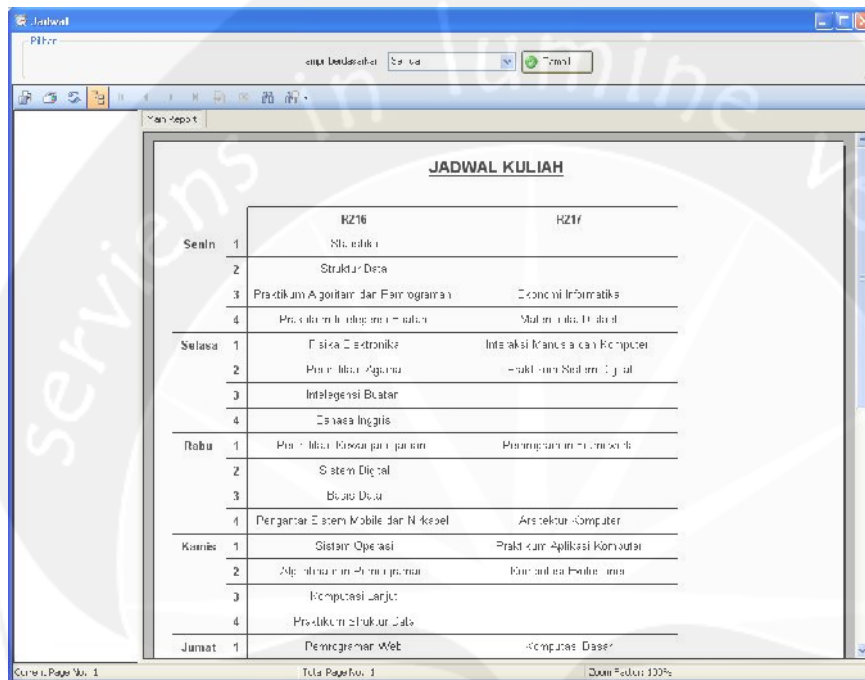
#### **On Click Button Cari**

Akan dilakukan pengecekan apakah kriteria pencarian telah dipilih. Jika sudah, maka akan dilakukan pencarian didalam tabel jadwal berdasarkan kriteria dan kata kunci pencarian yang telah diisikan oleh pengguna. Jika belum maka akan ditampilkan pesan “pilih kriteria pencarian terlebih dahulu”.

*Query* yang digunakan:

“select id\_jadwal, id\_mk, sesi, id\_ruang, id\_dosen, semester from jadwal\_mk where " + kategori\_pencarian + " like '%" + kata\_kunci + "%'”.

Perancangan antarmuka untuk menampilkan jadwal yang terbangkitkan diperlukan saat user hendak melihat hasil pembangkitannya. Antarmuka ini digunakan oleh pengguna untuk melihat jadwal yang telah tersusun dan mencetak jadwal tersebut.



		RZ16	RZ17
Senin	1	Statistika	
	2	Struktur Data	
	3	Praktikum Algoritma dan Pemrograman	Etika Informatika
	4	Prosedur Pengembangan Sistem	Matematika Diskrit
Selasa	1	Fisika Elektronika	Interaksi Manusia dan Komputer
	2	Penerbitan Aplikasi	Kejuruan Sistem Digital
	3	Inteligensi Buatan	
	4	Bahasa Inggris	
Rabu	1	Penerbitan Pengembangan Sistem	Pengantar Sistem Mobile
	2	Sistem Digital	
	3	Bahasa Inggris	
	4	Pengantar Sistem Mobile dan Nirkabel	Arsitektur Komputer
Kamis	1	Sistem Operasi	Praktikum Aplikasi Komputer
	2	Algoritma dan Pemrograman	Kendaraan Bermotor
	3	Komputasi Lanjut	
	4	Produksi Struktur Data	
Jumat	1	Pengantar Web	Komputer Dasar

Gambar 9 Antarmuka Cetak Jadwal

Fungsi yang dapat dilakukan melalui antarmuka ini adalah: tampil jadwal dan cetak jadwal.

Tampil jadwal digunakan untuk menampilkan jadwal mata kuliah yang telah tersimpan di dalam basis data ke dalam jendela report dan cetak jadwal digunakan untuk mencetak jadwal mata kuliah yang telah ditampilkan pada jendela report.

Data di dalam combobox harus dipilih terlebih dahulu sebelum melakukan tampil jadwal dan cetak jadwal.



### **On Form Load**

Jendela pada report akan kosong, dan data nama dosen yang mengajar akan dimasukkan ke dalam combobox.

### **On Click Button Tampil**

Akan ditampilkan jadwal mata kuliah ke dalam jendela report apabila combobox kriteria penampilan telah dipilih. Apabila kriteria penampilan belum dipilih akan ditampilkan pesan “pilih kriteria terlebih dahulu”. Kemudian apabila belum terdapat jadwal di dalam basis data, maka akan ditampilkan pesan “Jadwal mata kuliah masih kosong”.

*Query* yang digunakan:

“select \* from jadwal\_mk where id\_ruang is not null”.

atau

“select \* from jadwal\_mk where id\_dosen= ? and id\_ruang is not null”.

### **On Click Button Cetak**

Akan dilakukan pencetak data sesuai data yang tertampil pada jendela report.

Perancangan Hitory merupakan fasilitas yang dipergunakan untuk menyimpan semua jadwal yang pernah dibangkitkan. Antarmuka ini digunakan oleh pengguna untuk melihat jadwal yang pernah tersusun menggunakan sistem ini.

Id	Index	Ir MK	Har	Sesi	Id Ruang	Id Dosen	Tanggal
1	1	INF1323	Senn	2	R215	5	22/02/2010
2	1	MPK03	Selesa	2	R215	5	22/02/2010
3	1	INF1223	Robu	2	R215	5	22/02/2010
4	1	INF1313	Kemis	2	R215	1	22/02/2010
5	1	INF3412	Jumat	2	R215	1	22/02/2010
6	1	INF3711	Senn	3	R215	1	22/02/2010
7	1	INF1633	Selesa	3	R215	5	22/02/2010
8	1	INF1634	Habu	4	R215	4	22/02/2010
9	1	INF1533	Kemis	3	R215	5	22/02/2010
10	1	INF3714	Jumat	3	R215	3	22/02/2010
11	1	INF1023	Senn	1	R215	5	22/02/2010
12	1	INF3312	Selesa	1	R215	1	22/02/2010
13	1	MPK0203	Robu	1	R215	1	22/02/2010
14	1	INF2233	Kemis	1	R215	5	22/02/2010
15	1	INF2332	Jumat	1	R215	1	22/02/2010
16	1	INF1731	Senn	4	R215	1	22/02/2010
17	1	MPK0402	Selesa	4	R215	5	22/02/2010
18	1	INF2842	Robu	4	R215	1	22/02/2010

Gambar 10 Antarmuka History

Fungsi yang dapat dilakukan melalui antarmuka ini adalah: tampil history dan cari history.

Tampil history dilakukan ketika pertama kali pengguna menggunakan antarmuka tersebut, dimana data history yang ada di dalam basis data akan ditampilkan ke dalam tabel history. Kemudian cari history digunakan untuk mencari data history di dalam basis data.

#### On Form Load

Data history yang berada di dalam basis data akan ditampilkan kedalam tabel history.

Query yang digunakan:

“select id\_histoty, index\_hasil, id\_mk, hari, sensi, id\_ruang, id\_dosen, tanggal from history order by id\_history”.

#### On Click Button Cari

Akan dilakukan pengecekan apakah kriteria pencarian telah dipilih. Jika sudah, maka akan dilakukan pencarian didalam tabel jadwal berdasarkan kriteria dan kata kunci pencarian yang telah diisikan oleh pengguna. Jika

belum maka akan ditampilkan pesan “pilih kriteria pencarian terlebih dahulu”.

*Query* yang digunakan:

```
“select id_history, index_hasil, id_mk, hari, sesi, id_ruang, id_dosen, tanggal  
from history where " + kategori_pencarian + " like '%" + kata_kunci + "%'”.
```



## BAB V

### HASIL DAN PEMBAHASAN

Setelah diuji, perangkat lunak menunjukkan kinerja dengan memberikan hasil yang dapat dideskripsi sebagai berikut.

Pengujian fungsional menunjukkan perangkat lunak berjalan benar. Tabel 1 Pengujian dapat ditunjukkan di bawah ini:

Tabel 1. Pengujian Fungsional Pembangkitan Jadwal Kuliah

Deskripsi	Pengujian pembangkitan jadwal oleh user.			
Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria hasil evaluasi	Hasil yang didapat
<ul style="list-style-type: none"> <li>Pilih mata kuliah yang akan disusun menjadi jadwal</li> <li>Pilih tombol generate.</li> </ul>	<ul style="list-style-type: none"> <li>Mata kuliah ; “Kalkulus”, “Struktur Data”, “Sistem Operasi”, “Sistem Informasi”</li> </ul>	<ul style="list-style-type: none"> <li>Jadwal berhasil disusun</li> <li>Tampilkan pesan “selesai”.</li> </ul>	<ul style="list-style-type: none"> <li>Jadwal berhasil disusun</li> <li>Keluar pesan “selesai”.</li> </ul>	<ul style="list-style-type: none"> <li>Jadwal berhasil disusun</li> <li>Pesan “selesai”.</li> </ul>
<b>Kesimpulan</b>	Handal			

Pengujian Penyimpanan hasil jadwal kuliah, ditunjukkan pada tabel 2 di bawah ini:

Tabel 2. Pengujian Fungsional Penyimpanan Hasil Jadwal Kuliah

Deskripsi	Pengujian simpan data jadwal oleh user.			
Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Kriteria hasil evaluasi	Hasil yang didapat
<ul style="list-style-type: none"> <li>Pilih tombol</li> </ul>		<ul style="list-style-type: none"> <li>Jadwal yang</li> </ul>	<ul style="list-style-type: none"> <li>Jadwal yang</li> </ul>	<ul style="list-style-type: none"> <li>Jadwal</li> </ul>

Simpan.		tersusun berhasil disimpan ke dalam basis data. – Tampilkan pesan “Simpan data berhasil”.	tersusun berhasil disimpan ke dalam basis data. – Keluar pesan “Simpan data berhasil”.	yang tersusun berhasil disimpan ke dalam basis data. – Pesan “Simpan data berhasil”.
<b>Kesimpulan</b>	Handal			

Pengujian fungsional menampilkan hasil jadwal kuliah ditunjukkan tabel 3.

Tabel 3. Pengujian Fungsional Proses Menampilkan Hasil Jadwal Kuliah

<b>Identifikasi</b>	<b>AUT-006-03</b>			
<b>Deskripsi</b>	Pengujian tampil data jadwal oleh user.			
<b>Prosedur Pengujian</b>	<b>Masukan</b>	<b>Keluaran yang diharapkan</b>	<b>Kriteria hasil evaluasi</b>	<b>Hasil yang didapat</b>
– Pilih menu mata kuliah		– Semua data mata kuliah berhasil ditampilkan	– Semua data mata kuliah berhasil ditampilkan	– Semua data mata kuliah berhasil ditampilkan
<b>Kesimpulan</b>	Handal			

Hasil menunjukkan bahwa terdapat jadwal optimal yang dibangkitkan dengan kendala ringan karena sumber daya yang berlebih. Hal ini menunjukkan semakin banyak sumber daya, semakin mudah dan semakin banyak alternatifnya jadwal dibuat. Namun demikian untuk preferensi tertentu yang digabung dengan sumber daya terbatas (tidak kurang) maka jadwal yang dibangkitkan ada yang tidak optimal. Semakin banyak kendala diterapkan semakin mungkin jadwal yang dibangkitkan tidak optimal. Pengulangan eksekusi aplikasi dapat memperbaiki solusi, yang tidak optimal menjadi optimal. Namun demikian hal ini tidak ada jaminannya.

Dalam kasus uji menunjukkan jadwal kuliah dengan preferensi menghindari sesi 5 untuk semua hari. Dibutuhkan dua ruang kelas untuk menyelenggarakan semua kuliah. Lihat tabel 4. Jadwal ini terkategori optimal.

Tabel 4. Contoh eksekusi dengan preferensi non-sesi 5.

Jadwal Kuliah			
Hari	Sesi	Ruang 3216	Ruang 3217
Senin	1	Logika Matematika, A, PA, 1	Data Mining, A, ERN, 7
	2	Sistem Informasi, A, AJS, 4	Pemrograman Jaringan, A, KA, 3
	3	Statistik, B, PA, 2	Algoritma dan Pemrog, C, IW, 1
	4	Sistem Digital, A, EDJ, 2	Basisdata, A, IW, 4
Selasa	1	Peng. Tekn Informasi, A, KA, 1	Analisis dan Per Jaringan. A, YSP, 7
	2	Algoritma dan Pemrog, B, PM, 1	Intelegensi Buatan, A, SYT, 3
	3	Statistik, A, ERN, 2	Fisika Elektronik, A, EDJ, 1
	4	Kalkulus, A, FSP, 1	Data Struktur Lanjut, A, PM, 3
Rabu	1	MSIK, A, YSP, 7	Basis Data, C, EDU, 4
	2	Algoritma dan Pemrog, A, EDU, 1	Ekonomi Informatika, A, YSP, 5
	3	Kalkulus, C, PA, 1	Simulasi dan Pemodelan, A, SYT, 7
	4	Grafika Komputer, A, SYT, 5	Kompjutsi Lanjut C, PRN, 3
Kamis	1	Kamutasi Lanjut, A, FSP, 3	Basisdata, B, IW, 4
	2	Algoritma dan Pemrog, D, FSR, 1	Strukturdata, A, EDU, 2
	3	Sistem Pend Keputusan, A, PM, 7	Arsitektur Komputer, A, EDJ, 3
	4	Pengolahan Citra, A, BYD, 7	Peng. Sistem Mobile & Wireless, A, TS, 4
Jumat	1	Logika Matematika, B, ERN, 1	Sistem Operasi, A, KA, 3
	2	PBO, A, BLS, 4	Teknik Optimasi, A, FSP, 7
	3	Penelitian Operasi, A, PM, 7	Kalkulus, B, PA, 1
	4	Perenc Sistem Inf Strategik, A, BLS, 7	Komputasi Lanjut, B, AJS, 3

Tidak optimalnya jadwal lebih disebabkan karena gagalnya dipenuhinya (1) kebutuhan alami jadwal (sumber daya) dan (2) preferensi yang terlalu banyak, yang kemungkinan memang tidak ada solusinya. Sayangnya PSO tidak dapat membuktikan bahwa solusi semacam itu tidak eksis.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1. Kesimpulan**

Berdasarkan pembahasan pada bab-bab sebelumnya, dapat ditarik kesimpulan :

1. PSO dengan metode berbasis populasinya telah berhasil membangkitkan jadwal kuliah.
2. Jadwal kuliah dapat dipandang sebagai masalah optimalisasi, sehingga dapat ditempuh dengan metode-metode optimalisasi lainnya.
3. Kecepatan penemuan jadwal kuliah relatif dapat membantu pejabat akademik dengan paling tidak alternatif jadwal kuliah.

#### **6.2. Saran**

Beberapa saran yang dapat ditarik dari proses analisa sampai pembuatan laporan penelitian ini adalah sebagai berikut :

1. Konvergensi PSO dalam metode yang dipakai penulis masih kurang bagus, disarankan ditempuh dengan alternatif metode lain di bawah PSO, seperti local best, atau modifikasi PSO lainnya.
2. Penambahan preferensi dapat ditambahkan, meski harus tetap disadari bahwa kadang tidak tercapainya keadaan optimal, bukan algoritmanya saja tetapi juga memang solusi semacam itu tidak pernah ada. Preferensi yang disarankan ditambahkan adalah, banyak dosen mengajar dalam sehari tidak lebih dari angka tertentu, dan dalam satu minggu tidak kurang dalam n hari mengajar.

## DAFTAR PUSTAKA

- [1] Adriane Mieke et al, "Tackling the University Course Timetabling Problem with an Aggregation Approach", PATAT06 Proceedings, 2006
- [2] Ai The Jin & Kachitvichyanukul Voratas, "A Particle Swarm Optimization for the Vehicle Routing Problem with Simultaneous Pickup and Deleivery", Computers and Operations Research Elsevier Journal, 2009.
- [3] Clerc Maurice, Particle Swarm Optimization, ISTE Ltd, 2006.
- [4] Dewi Findra Kartika Sari, Schedule Generation Using Coloring Graph, Thesis, UAJY, Yogyakarta, 2006.
- [5] Engelbrecht Andries P, Computational Intelligence An Introduction, Second Edition, John Wiley and Sons Ltd, 2007.
- [6] Hu, Xiaoui, 2005, *Particle Swarm Optimization*,  
<http://www.swarmintelligence.org/index.php>, diakses pada bulan Juli 2009.
- [7] Nadia, Knowledge Discovery In Academic Data With Association Rules Using C #, Thesis, UAJY, Yogyakarta, 2005.
- [8] Tsou, Ching-Shih et al, "Using Croeding Distance to Improve Multi Objective PSO with Local Search", Swarm Intelligence: Focus on Ant and Particle Swarm Optimization Proceedings, 2007